

In-voxel Neural Tract Orientation Statistics

Project Report

Yinpeng Li, Martin Styner¹

Abstract: We describe a new approach for estimating the underlying fiber configuration at each voxel, including the number of fibers passing through the voxel, the axial direction of each fiber, and the angular spread of each fiber. Our approach uses spherical k-means clustering and GAP analysis to perform statistics on the orientations of fiber segments inside the voxel, and transforms the statistical result into our estimation of the fiber configuration. In order to accelerate the clustering process, a subdivided icosahedron is used to discretize a unit sphere, on which the orientations are clustered. The strengths and limitations of this approach are demonstrated through comprehensive simulation experiments.

1. Introduction

Due to the anisotropic diffusion of water in organized tissues such as brain white matter, diffusion tensor imaging (DTI) and DTI-based fiber tractography have been employed to perform non-invasive investigation of neural architecture (Mukherjee, P., Berman, J., et al., 2008). Neuroscientists use these techniques to find out how neurons originating from one region connect to other regions, and how strong the connections might be.

Traditional single-tensor tractography method suffers from crossing fiber issues and limited SNR (Mori, S., van Zijl, P., 2002). Much has been done in the literature to address these problems. Among all the endeavors, Rathi et al. modeled the diffusion signal with weighted mixture of multiple Gaussian tensors, and the estimation of the model was performed by an unscented Kalman filter, using the estimation at previous positions as a guide (Malcolm, J., Rathi, Y., et al., 2009) (Lienhard, S., et al., 2011).

In order to get more accurate estimations of the fiber configuration at each voxel, we propose a new approach based on the results of existing tractography methods. We create large amount of seeds in each voxel and generate relatively short tracts with a small step length, so that the error accumulated during the tractography propagation process is minimized (Lazar, M., Alexander, A., 2003). After the tracts have been generated, we look into each voxel and do statistics on the orientations of fiber segments (Fig 1) that are inside of the voxel. This way, not only do we have local information of the current voxel, but we also have information of nearby voxels due to the fibers coming from them. The statistical result of these orientations will reflect the underlying fiber configuration, such as the number of fibers passing through the voxel, the axial direction of each fiber, and the angular spread of each fiber. The result of our method could be further used in ODF computation and connectivity analysis.

¹ Also the supervisor of this project.

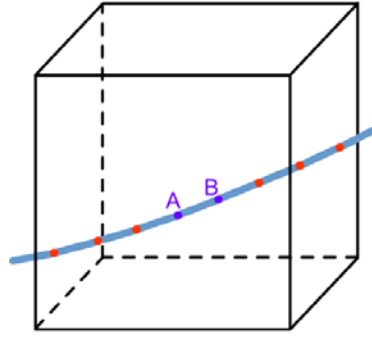


Fig 1 A tract passing through a voxel. The ends of segments are marked with red and purple dots.
Note that for segment AB, both \overrightarrow{AB} and \overrightarrow{BA} are its orientation vectors.

This project focuses on the orientation statistics part of the process. A method to perform the statistics is implemented, and its strengths and limitations are demonstrated through comprehensive simulation experiments.

2. Methodology

The implementation of the orientation statistics is based on spherical k-means clustering (Dhillon, I., Modha, D., 2001) and GAP analysis (Tibshirani, R., Walther G., et al., 2001).

2.1 Sphere discretization

To perform the orientation statistics, we first fit a unit sphere at the center of the voxel, and transform all the fiber segment orientation vectors to unit vectors whose tails coincide with the center of the unit sphere. Thus, all the heads of these vectors lie on the surface of the unit sphere, and we can apply spherical k-means clustering algorithm on them.

In addition, to expedite the computation, we use an iteratively subdivided icosahedron model (Fig 2) to discretize the unit sphere. We fit an inscribed subdivided icosahedron into the unit sphere, and associate a bin at each of its vertices. This way, every orientation vector pierces one triangle of the subdivided icosahedron, and the weight of the orientation vector (which is 1.0) is distributed among the bins pertaining to the three vertices of the triangle according to the barycentric coordinates of the intersection point (Fig 3). After all the orientation vectors have been pre-processed in this way, we have the orientation distribution info discretized into the bins associated to the vertices of the subdivided icosahedron, and we can perform statistics on these vertices to get the result we want.

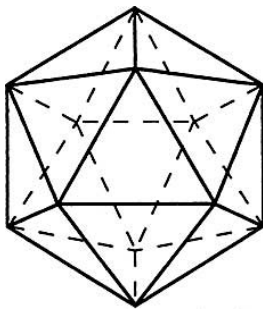


Fig 2 An icosahedron with subdivision level of 0.

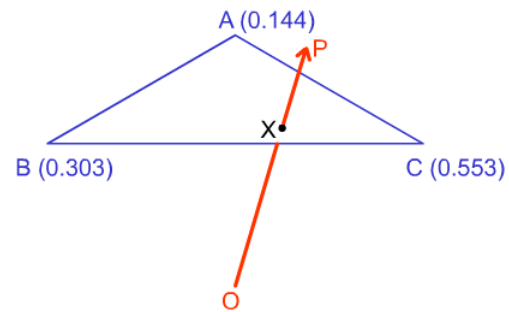


Fig 3 An orientation vector \overrightarrow{OP} pierces an equilateral triangle $\triangle ABC$. The number beside each vertex is the corresponding barycentric coordinate of the intersection point X, namely $\mathbf{X} = 0.144\mathbf{A} + 0.303\mathbf{B} + 0.553\mathbf{C}$.

2.2 Spherical k-means clustering

Traditional spherical k-means clustering algorithm is implemented in this project. To make the illustration more clear, we start with some necessary notation. Let N be the number of vertices of the subdivided icosahedron, O be the center of the unit sphere (also the center of the subdivided icosahedron), and let $\mathbf{P} = \{P_0, P_1, \dots, P_{N-1}\}$ denote the set of vertices. Also let $w(P_i)$ equal the value of the bin associated to vertex P_i .

The overall flow of the algorithm is as follows:

- i. Preprocessing: Rule out all the vertices P_i such that

$$\frac{w(P_i)}{\sum_{i=0}^{N-1} w(P_i)} < \text{THRESHOLD}$$

This is intended to filter out white noise in the data. The THRESHOLD is set to 0.001 in this project. Only the vertices that pass this test can enter the next stage of computation, and the set of these vertices is denoted as \mathbf{P}' .

- ii. Initialization: Create an initial partitioning. Multiple initialization strategies have been implemented in this project, including random partitioning, partitioning based on modulus, and evenly partitioning the list of vertices into K parts, which takes advantage of the fact that all the additional vertices created by subdividing a triangle of a level-0 icosahedron are listed contiguously in the icosahedron vertex list. The last strategy is implementation-dependent, but proves to yield best results.
- iii. Centroid estimation: For cluster j , let \mathbf{S}_j denote the set of vertices in cluster j , and compute the new centroid of cluster j as

$$\bar{\mu}_j = \frac{\sum_{X \in \mathbf{S}_j} \overrightarrow{OX}}{\|\sum_{X \in \mathbf{S}_j} \overrightarrow{OX}\|}$$

- iv. Data assignment: For each vertex $X \in \mathbf{P}'$, find the centroid closest in cosine similarity to \overrightarrow{OX} , and move X to the cluster which the centroid belongs to, if necessary.
- v. Stop if the partitioning does not change in this iteration. Otherwise go back to step iii.

2.3 GAP analysis

In this project, we focus on differentiating among the following three cases: $K = 2$ (single-fiber configuration), $K = 4$ (double-fiber configuration) and $K = 6$ (triple-fiber configuration). The isotropic case ($K = 0$) can easily be dealt with by performing a χ^2 test on the values of bins against the uniform distribution beforehand, and is thus excluded from our consideration. We use GAP analysis to pick the correct number of clusters in the data.

We first perform preprocessing on the results of spherical k-means clustering. For each member of the cluster set we obtain with a specific K , we compute its on-sphere-center M_j by minimizing

$$\sum_{X \in \mathbf{S}_j} w(X) \times \text{dos}(\text{stc}(\varphi, \theta), X)^2$$

with the Levenberg-Marquardt algorithm over the spherical coordinate space. Here, $w(X)$ denotes the value of bin associated to X , \mathbf{S}_j denotes the set of vertices in cluster j , function $\text{dos}()$ computes the on-sphere distance between two points on the unit sphere, function $\text{stc}()$ performs the spherical-to-Cartesian coordinate transformation, and (φ, θ) is the spherical coordinates of M_j . The Levenberg-Marquardt algorithm is initialized with

$$(\varphi_0, \theta_0) = \text{cts}(\mathbf{O} + \overline{\mu_j})$$

where $\overline{\mu_j}$ is the centroid of cluster j , and function $\text{cts}()$ performs the Cartesian-to-spherical coordinate transformation.

We choose uniform distribution as the reference distribution in the GAP analysis, and the following error measures have been implemented:

$$\begin{aligned} 1) \quad W_K &= \frac{\sum_{j=0}^{K-1} \sum_{X \in \mathbf{S}_j} w(X) \times (1 - \overline{\mathbf{O}\mathbf{X}} \cdot \overline{\mu_j})}{\sum_{X \in \mathbf{S}_j} w(X)} & 2) \quad W_K &= \frac{\sum_{j=0}^{K-1} \sum_{X \in \mathbf{S}_j} w(X) \times \text{dos}(\mathbf{M}_j, \mathbf{X})^2}{\sum_{X \in \mathbf{S}_j} w(X)} & 3) \quad W_K &= \frac{\sum_{j=0}^{K-1} \sum_{X, Y \in \mathbf{S}_j} w(X) \times w(Y) \times \text{dos}(X, Y)^2}{2 \times \sum_{X \in \mathbf{S}_j} w(X)} \end{aligned}$$

The last two error measures prove to give performance on par with each other, and both of them are stronger than the first. Thus, we use the last error measure in our GAP implementation.

The overall flow of the algorithm is as follows:

For each $K \in \{2, 4, 6\}$, we compute $\text{Gap}(K) = \log(W_K^*) - \log(W_K)$, where W_K^* is the error measure computed for the reference distribution. The K which yields the largest $\text{Gap}(K)$ is picked as the correct number of clusters.

After the K is picked, the cluster set computed by the spherical k-means clustering algorithm with the selected K is returned as the result of our method.

3. Evaluations

The method described above is evaluated through comprehensive simulation experiments. In each experiment, data is generated by simulating the specified ground-truth fiber configuration, and the result of our method is compared with this specification.

The specification of a ground-truth fiber configuration in a voxel includes the following items:

- i. Number of fibers passing through the voxel.
- ii. The axial direction of each fiber.
- iii. The angular spread of each fiber, specified by the standard deviation in angle domain.
- iv. The number (population) of sample directions that should be generated for each fiber.
- v. The number (population) of sample directions that should be generated as white noise.

When generating the simulation data for a configuration consisting of M fibers, we are actually generating M pairs of clusters (namely $2 \times M$ clusters in total), and the two clusters in each pair are symmetric with respect to the center of the unit sphere. This is due to the fact that in practice, for every fiber segment AB , both \overrightarrow{AB} and \overrightarrow{BA} count as its orientation vectors, thus the data we deal with will always be symmetric (Fig 4).

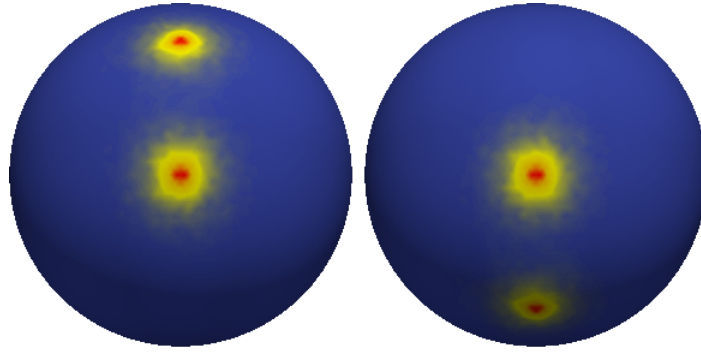


Fig 4 The simulation data generated for a two-fiber configuration, visualized by ParaView². The axial directions of the two fibers are $(1, 0, 0)$ and $(0.707, 0, 0.707)$ respectively, namely the two fibers form an angle of 45° . The angular spread of both fibers is 10° , and the sample population for each fiber is 5000. The population of noise samples is set to 0. The left picture is viewed in $-X$ direction, and the right one is viewed in $+X$ direction, both taken with $+Z$ pointing upwards.

3.1 Criterion for success

We use the following criterion to determine whether our method has successfully estimated the ground-truth fiber configuration in the voxel:

The K clusters we get with our method form $K/2$ pairs of clusters, and $K/2$ equals the number of fibers in the configuration specification. The axial direction of each cluster should not deviate from the corresponding fiber's axial direction by more than 5° .

Here, the axial direction of a cluster j is defined as $\overline{OM_j}$, where M_j is the on-sphere-center of the cluster computed in the preprocessing step of GAP analysis.

3.2 Assumptions

Since we use a subdivided icosahedron to discretize the unit sphere, it's self-evident that the higher subdivision level we use, the more accurate result we get (Fig 5). Nevertheless, as the subdivision level increases, so does the computation time for a single voxel (Fig 6).

² ParaView: A data visualization application. <http://www.paraview.org/>

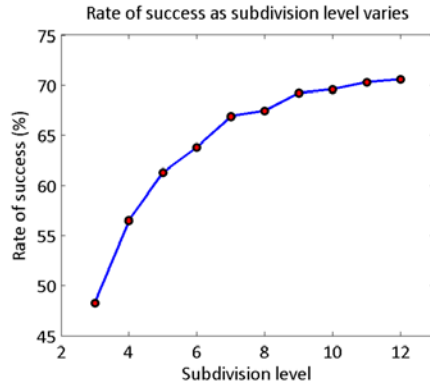


Fig 5 The rate of successfully estimating the ground-truth fiber configurations when the algorithm is run on a set of 1296 two-fiber test cases with subdivision level varying from 3 to 12. The test cases are generated by varying the parameters of double-fiber configurations. Among all the parameters, the ratio between the sample populations of two fibers varies from 1:1 to 1:10, the angular spread of the two fibers varies from 5° to 20°, and the separation angle between the two fibers varies from 5° to 90°. The sample population for white noise is fixed at 0.

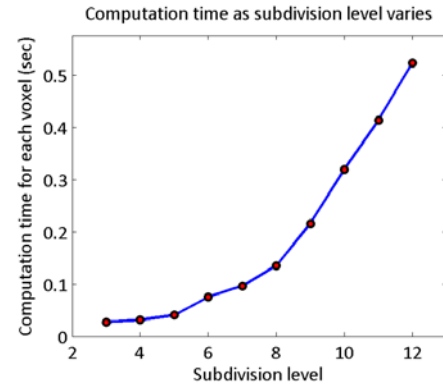


Fig 6 The average time of computation spent on a single voxel with different subdivision level used. The time is measured within a single-thread process run on a Dual-Core AMD Opteron(tm) Processor 8220 machine.

In the remaining part of this report, we present the results of experiments with subdivision level fixed at 7.

Besides, we also make the following assumptions during the simulation experiments:

- i. All the fibers passing through the same voxel have the same angular spread.
- ii. The sample population for white noise cannot exceed the sum of sample population for all the fibers.

3.3 Results for single-fiber configurations

The variable parameters for the simulations of single-fiber configurations include the population of fiber samples, the population of white noise samples, and the angular spread of the fiber. The parameter groups we use in simulations are listed in Table 1.

Table 1 Parameter groups used in single-fiber simulations

Population of fiber samples	Population of white noise samples	Angular spread (°)
1000	0 ... 1000, with step 100	3 ... 30, with step 3
2000	0 ... 2000, with step 200	3 ... 30, with step 3
3000	0 ... 3000, with step 300	3 ... 30, with step 3
4000	0 ... 4000, with step 400	3 ... 30, with step 3
5000	0 ... 5000, with step 500	3 ... 30, with step 3
6000	0 ... 6000, with step 600	3 ... 30, with step 3
7000	0 ... 7000, with step 700	3 ... 30, with step 3
8000	0 ... 8000, with step 800	3 ... 30, with step 3
9000	0 ... 9000, with step 900	3 ... 30, with step 3
10000	0 ... 10000, with step 1000	3 ... 30, with step 3

Our algorithm successfully estimates the underlying fiber configurations for all the test cases generated with the parameter groups in the table.

We define the error of our result as the maximum deviation (in degrees) between the axial direction of any cluster and the cluster's corresponding fiber's axial direction. We find through simulation experiments that for single-fiber configurations:

- The error decreases as the fiber sample population increases (Fig 7).
- The error increases as the fiber angular spread increases (Fig 8).
- The noise sample population has little effect on the error (Fig 9).

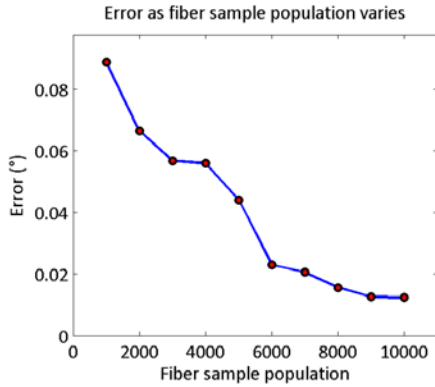


Fig 7 Error as the fiber sample population varies, whereas the noise sample population is fixed at 0, and the angular spread is fixed at 3°.

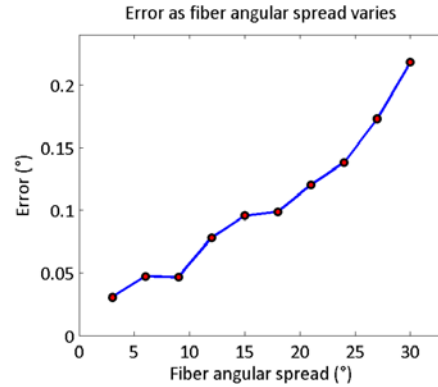


Fig 8 Error as the angular spread varies, whereas the noise sample population is fixed at 0, and the fiber sample population is fixed at 10000.

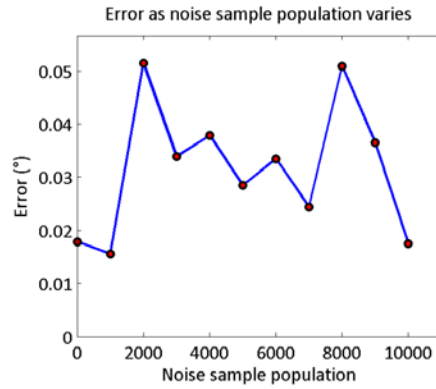


Fig 9 Error as the noise sample population varies, whereas the fiber sample population is fixed at 10000, and the angular spread is fixed at 3°.

3.4 Results for double-fiber configurations

The variable parameters for the simulations of double-fiber configurations include the population of fiber samples for each fiber, the population of white noise samples, the angular spread of the two fibers, and the angular separation between the two fibers. The parameter groups we use in simulations are listed in Table 2.

Table 2 Parameter groups used in double-fiber simulations

Population of fiber samples ³		Population of noise samples	Angular spread (°)	Angular separation (°)
Fiber1	Fiber2			
5000	5000	0 ... 10000, with step 1000	3 ... 30, with step 3	3 ... 90, with step 3
4500	5000	0 ... 9500, with step 950	3 ... 30, with step 3	3 ... 90, with step 3

³ Without loss of generality, we assume in the table that Fiber2 always has a sample population larger than or equal to the sample population of Fiber1.

Population of fiber samples ³		Population of noise samples	Angular spread (°)	Angular separation (°)
4000	5000	0 ... 9000, with step 900	3 ... 30, with step 3	3 ... 90, with step 3
3500	5000	0 ... 8500, with step 850	3 ... 30, with step 3	3 ... 90, with step 3
3000	5000	0 ... 8000, with step 800	3 ... 30, with step 3	3 ... 90, with step 3
2500	5000	0 ... 7500, with step 750	3 ... 30, with step 3	3 ... 90, with step 3
2000	5000	0 ... 7000, with step 700	3 ... 30, with step 3	3 ... 90, with step 3
1500	5000	0 ... 6500, with step 650	3 ... 30, with step 3	3 ... 90, with step 3
1000	5000	0 ... 6000, with step 600	3 ... 30, with step 3	3 ... 90, with step 3
500	5000	0 ... 5500, with step 550	3 ... 30, with step 3	3 ... 90, with step 3

We find through simulation experiments that for double-fiber configurations:

- The rate of success drops almost quadratically as the ratio between the sample population of two fibers goes below 0.4 (Fig 10).
- The noise sample population has little effect on the result (Fig 10).
- Generally, the rate of success starts dropping sharply to 0 as the angular separation between the two fibers goes below 2σ , where σ is the angular spread of the two fibers (Fig 11).

Rate of success as fiber population ratio and noise/signal ratio vary

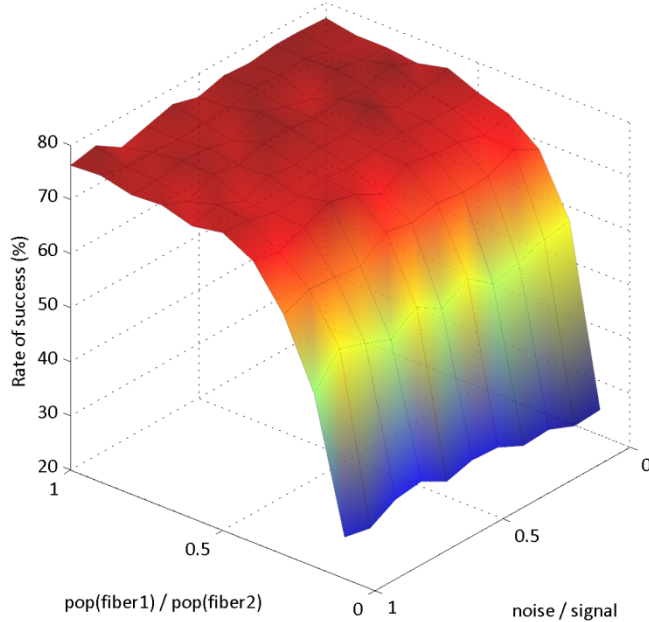


Fig 10 The rate of successfully estimating the ground-truth fiber configurations as the ratio between the sample population of two fibers and the noise/signal ratio vary. In the figure, pop(fiber1) denotes the sample population of fiber1, and we always assume that the population of fiber2 is larger. The noise/signal ratio is computed by equation $\frac{\text{pop}(\text{noise})}{\text{pop}(\text{fiber1}) + \text{pop}(\text{fiber2})}$.

Rate of success as angular separation and angular spread vary

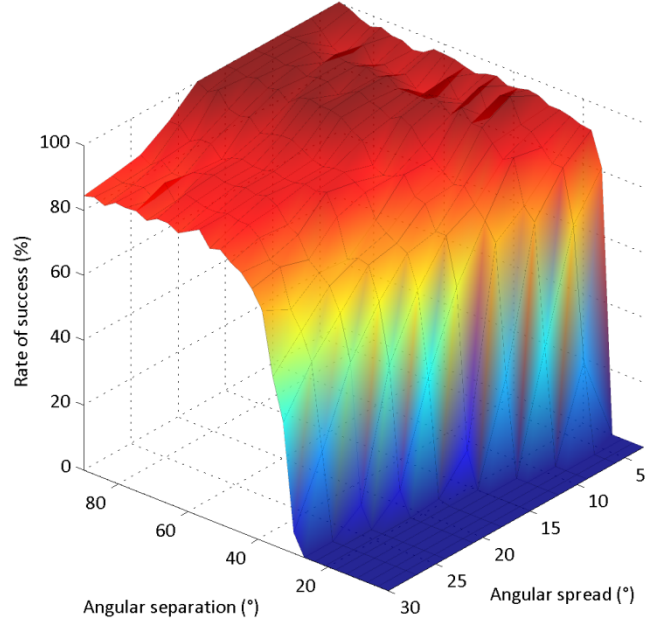


Fig 11 The rate of successfully estimating the ground-truth fiber configurations as the angular separation between the two fibers and the angular spread of the two fibers vary.

3.5 Results for triple-fiber configurations

Due to the fact that it's very difficult to properly enumerate all the triple-fiber configurations, no exhaustive simulation experiments are done for them. Only randomly generated test cases have been used in the simulation experiments, but the results are not general enough to be included in this report.

4. Discussion & Conclusion

At first, we tried to apply numerical methods (Levenberg-Marquardt, etc.) directly to fit weighted mixture of Gaussian distributions on the discretized orientation data. However, the instability of numerical methods rendered our idea impractical. In the end, we resorted to traditional machine learning methods to attack the problem.

Despite fast convergence, the k-means algorithm has two major limitations. The first limitation is that the algorithm tends to find clusters of comparable spatial extent, which is why we made the assumption that all the fibers passing through the same voxel have the same angular spread. The second limitation is that the k-means clustering algorithm is sensitive to initialization, which is also the major issue with our current method. The algorithm can easily get stuck in local minima and result in failure if not initialized properly. Although we can repeat the clustering process multiple times with random initializations, the probability is still not on our side. Hierarchical clustering methods, such as bisecting k-means and STING, will be able to address this issue better, but the time cost by these methods might be unaffordable for a voxel-by-voxel computation application.

All in all, even under our assumptions (listed in section 3.2), the current method is far from satisfying. The method works well for single-fiber configurations and double-fiber configurations with similar fiber sample populations, but its performance degrades drastically as the difference between populations of different fibers gets larger. In order to make this method usable in a medical application, either a better way of initializing the current k-means clustering algorithm must be found, or a fast enough hierarchical clustering method must be implemented instead.

5. References

- Dhillon, I., Modha, D. (2001). Concept Decompositions for Large Sparse Text Data using Clustering. *Machine Learning*.
- Lazar, M., Alexander, A. (2003). An error analysis of white matter tractography methods : synthetic diffusion tensor field simulations. *NeuroImage*.
- Lienhard, S., et al. (2011). A full bi-tensor neural tractography algorithm using the unscented Kalman filter. *EURASIP Journal on Advances in Signal Processing*.
- Malcolm, J., Rathi, Y., et al. (2009). Neural tractography using an unscented Kalman filter. *Inf Process Med Imaging*.
- Mori, S., van Zijl, P. (2002). Fiber tracking: principles and strategies - a technical review. *NMR in Biomedicine*.
- Mukherjee, P., Berman, J., et al. (2008, April). Diffusion Tensor MR Imaging and Fiber Tractography: Theoretic Underpinnings. *American Journal of Neuroradiology*.
- Tibshirani, R., Walther G., et al. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B*.

Appendix

The code for this project is hosted on NITRC, and can be checked out with svn at <https://www.nitrc.org/svn/fiberodf>.

The documentation for the code can be found at <http://www.nitrc.org/plugins/mwiki/index.php/fiberodf:MainPage>.